Algorithm:

1. The given URL is parsed using Soup library. We ignore non-alphanumeric characters, special characters and punctuation marks to prepare text representing the Title and Body.

2. This returned text in String form is used to create a HashMap of words and their frequency of occurrences.

3. This Hashmap is reversed and sorted for getting top N frequency words.

4. For Title: Each word occurrence is checked with the body text to decide if it is a keyword
For Body: Out of the top N frequency words a list of one/two-word phrase which can stand for page description is returned.

ParsingURL:

The ParsingURL class takes in URL given by the user, connect via HTTP using the Jsoup library to fetch the contents of the webpage, and construct a corpus of text that lives on that webpage. This corpus contains the title of the webpage, and all the text in the body.

We prune out non-alphanumeric characters because they are uninteresting in terms of getting any relevant information. We also prune words that have "'s" in the end. This is so that words like Dawson and Dawson's are treated in the same way.

WordCount:

The WordCount class performs operation on the content returned by ParsingURL in the form of String and returns a HashMap of <word,frequency> after ignoring some stop words mentioned in a Set(leaveWords) made by stopWordSet String array

HashMapOrderedSet:

The HashMapOrderedSet class sorts the returned HashMap entries from WordCount class and returns the top N frequencies.
It also returns reverse HashMap containing top N frequencies as Key and respective word-list as Value

KeywordAnalysis:

The KeywordAnalysis class decides which words or phrases are keywords defining the webpage

For body: It checks single and two-word phrases by calculating their density using the formula- Density = frequency/#totalWords*100 and compares it with a minimum density (= 1.0 in our case)
For Title: It decides which word in title should be a keyword after checking it's frequency in the body text